OPEN ACCESS Freely available online

PLOS ONE

# Reconstruction of Ancestral Gene Orders Using Probabilistic and Gene Encoding Approaches

**Ning Yang[3], Fei Hu[1,2], Lingxi Zhou[1,2], Jijun Tang[1,2]***

**1** Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin, China, **2** Department of Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, United States of America, **3** School of Automation, Northwestern Polytechnical University, Xi'an, shaanxi, China

## Abstract

Current tools used in the reconstruction of ancestral gene orders often fall into event-based and adjacency-based methods according to the principles they follow. Event-based methods such as GRAPPA are very accurate but with extremely high complexity, while more recent methods based on gene adjacencies such as InferCARsPro is relatively faster, but often produces an excessive number of chromosomes. This issue is mitigated by newer methods such as GapAdj, however it sacrifices a considerable portion of accuracy. We recently developed an adjacency-based method in the probabilistic framework called PMAG to infer ancestral gene orders. PMAG relies on calculating the conditional probabilities of gene adjacencies that are found in the leaf genomes using the Bayes' theorem. It uses a novel transition model which accounts for adjacency changes along the tree branches as well as a re-rooting procedure to prevent any information loss. In this paper, we improved PMAG with a new method to assemble gene adjacencies into valid gene orders, using an exact solver for traveling salesman problem (TSP) to maximize the overall conditional probabilities. We conducted a series of simulation experiments using a wide range of configurations. The first set of experiments was to verify the effectiveness of our strategy of using the better transition model and re-rooting the tree under the targeted ancestral genome. PMAG was then thoroughly compared in terms of three measurements with its four major competitors including InferCARsPro, GapAdj, GASTS and SCJ in order to assess their performances. According to the results, PMAG demonstrates superior performance in terms of adjacency, distance and assembly accuracies, and yet achieves comparable running time, even all TSP instances were solved exactly. PMAG is available for free at http://phylo.cse.sc.edu.

**Citation:** Yang N, Hu F, Zhou L, Tang J (2014) Reconstruction of Ancestral Gene Orders Using Probabilistic and Gene Encoding Approaches. PLoS ONE 9(10): e108796. doi:10.1371/journal.pone.0108796

**Editor:** Gabriel Moreno-Hagelsieb, Wilfrid Laurier University, Canada

**Received** March 4, 2014; **Accepted** September 3, 2014; **Published** October 10, 2014



**Data Availability:** The authors confirm that all data underlying the findings are fully available without restriction. All relevant data are within the paper and the Supporting Information files.

**Funding:** The authors were funded by NSF IIS 1161586 and an internal grant from Tianjin University, China. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

* Email: jtang@cse.sc.edu

## Introduction

### Overview

Evolutionary biologists have had a long tradition in reconstructing genomes of extinct ancestral species. Mutations in a genomic sequence are made up not only at the level of base-pair changes but also by rearrangement operations on chromosomal structures such as inversions, transpositions, fissions and fusions [2]. Over the past few years, ancestral gene-order inference has brought profound predictions of protein functional shift and positive selection [3].

Methods for ancestral genome reconstruction either assume a given phylogeny that represents the evolutionary history among given species or search the most appropriate tree along with a set of ancestral genomes to fit the observed data. Depending on how the gene-order data is interpreted and handled, methods for solving the latter can be partitioned into two groups: event-based methods and adjacency-based methods. Event-based methods typically search for the set of ancestral gene orders that minimizes the sum of rearrangement distances over all edges of the given phylogeny. However, methods seeking exact solutions following such paradigm (such as GRAPPA [4], MGR [5,6]) have already encountered huge difficulties in handling modern genomes due to their NP-hard complexity. In consequence, methods such as GASTS [7] were developed to provide heuristic solutions. In addition to event-based methods, a number of adjacency-based methods have been proposed such as SCJ [8], InferCARsPro [9] and GapAdj [10]. Instead of explicitly considering a predefined set of rearrangement events, these methods take gene adjacencies into account and treat them as binary characters with present and absence states. In this way, by viewing the gene order as a set of gene adjacencies, the goal is to determine which adjacencies are contained in the genome.

We recently developed an adjacency-based method in the probabilistic framework called PMAG [1] to reconstruct ancestral genomic orders given a phylogeny. In this paper, we improved PMAG to introduce a better algorithm that can assemble gene orders with fewer contigs, hence provided better accuracy. This new algorithm is also faster, enabling us to handle larger datasets. Through simulation experiments, we verify the usefulness of our biased transition model and re-rooting procedure we incorporate in the program. Then the performance of PMAG is evaluated against other existing methods including InferCARsPro, GapAdj, SCJ and GASTS under a wide range of settings.

## Genome rearrangement

Given a set of $n$ genes $\{g_1,g_2,\cdots,g_n\}$, a genome can be represented by an *ordering* of these genes. To indicate the strandedness of genes, each gene is assigned with an orientation that is either positive, written $g_i$, or negative, written $-g_i$. Two genes $i$ and $j$ are said to be *adjacent* in genome $G$ if $i$ is immediately followed by $j$, or, equivalently, $-j$ is immediately followed by $-i$. A *breakpoint* of two genomes is defined as an adjacency appears in one but not in the other.

Let $G$ be the multi-chromosomal genome with signed ordering $\{a_1,a_2,\cdots,a_n\}$, $\{b_1,b_2,\cdots,b_m\},\cdots$ ($\{\cdots\}$indicates a chromosome). An *inversion* (also called *reversal*) between indices $i$ and $j$ ($i\leq j$) of chromosome $a$, produces a chromosome $a'$ with linear ordering

$$a_1,a_2,\cdots,a_{i-1},-a_j,-a_{j-1},\cdots,-a_i,a_{j+1},\cdots,a_n.$$

For example, the identity genome $\{1,2,3,4,5,6\}$ is transformed into $\{1,-4,-3,-2,5,6\}$ when the gene block $\{2,3,4\}$ is inverted.

A *transposition* on a chromosome $a$ acts on three indices $i,j,k$, with $i\leq j$ and $k\notin[i,j]$, picking up the interval $a_i,\cdots,a_j$ and inserting it immediately after $a_k$. Thus the chromosome $a$ of the genome is replaced by (assume $k>j$):

$$a_1,\cdots,a_{i-1},a_{j+1},\cdots,a_k,a_i,a_{i+1},\cdots,a_j,a_{k+1},\cdots,a_n.$$

For example, genome $\{1,2,3,4,5,6\}$ is transformed into $\{1,5,2,3,4,6\}$ when the gene block $\{2,3,4\}$ is moved in front of gene 6.

A *translocation* on a genome $G$ acts on two chromosomes $a=\{a_1,a_2,\cdots,a_n\}$ and $b=\{b_1,b_2,\cdots,b_m\}$. Given two indices $i,j$, it picks up the interval $a_i,\cdots,a_n$ and $b_j,\cdots,b_m$ and then changes their places. Thus the two chromosomes $a,b$ of genome $G$ become

$$\{a_1,a_2,\cdots,a_{i-1},b_j,\cdots,b_m\},\{b_1,b_2,\cdots,b_{j-1},a_i,\cdots,a_n\}$$

Yancopoulos [11] proposed a universal double-cut-and-join operation that accounts for inversions, transposition and translocations which resulted in a new genomic distance that can be computed in linear time. In particular, a DCJ operation consists of cutting two connections (breakpoints) in the genome, and rejoining the resulting four unconnected ends in two new pairs. Although there is no direct biological evidence for DCJ operations, these operations are very attractive because they provide a simpler and unifying model for genome rearrangement.

Later the Single-Cut-or-Join (SCJ) [12] operation was proposed as a basis for a new rearrangement distance between multi-chromosomal genomes, leading to very fast algorithms. The SCJ operation is modeled on the two most fundamental rearrangement operations—the cutting and joining of adjacencies. A cutting operation breaks an adjacency into two telomeres, and a joining operation is performed in the opposite way by pairing two telomeres into an adjacency. Any cutting or joining applied to the genome will be called a Single-Cut-or-Join (SCJ) operation. Since the genome is represented as a set of adjacencies, a cutting can also be viewed as the removal of an adjacency from the set while the joining is the addition of an adjacency.

Event-based methods typically iterate over each internal node to solve for the median genomes until the sum of rearrangement events over all edge distances (tree score) is minimized. The median problem can be formalized as follows: given a set of $m$ genomes with permutations $\{x_i\}_{1\leq i\leq m}$ and a distance measurement $d$, find another permutation $x_t$ such that the median score defined as $\sum_{i=1}^{m} d(x_i,x_t)$ is minimized. However solving even the simplest case when $m$ equals to three is NP-hard for most distance measurements [13,14]. Among all existing median solvers, the best is the DCJ median solver proposed by Xu and Sankoff (ASMedian [15]) based on the concept of adequate sub-graph and decomposes a multiple breakpoint graph [16] into smaller and easier graphs. Although ASMedian could remarkably scale down the computational expenses of median searching, it yet runs very slow when the genomes are distant. Based on ASMedian, Xu developed its heuristic algorithm GASTS that can quickly score a fixed phylogenetic tree and enables us to attack previously unapproachable problems by GRAPPA and MGR, as demonstrated on a set of vertebrate genome with over 2,000 genes.

## Ancestral gene-order reconstruction based on gene adjacencies

Over recent years, a collection of models based on the study of gene adjacencies have been proposed for solving various rearrangement problems. In these models, each gene adjacency is considered as a binary character and the problem of reconstructing ancestral genomes can then be reduced as deciding, for every adjacency, whether an ancestral genome contains the adjacency.

The breakpoint-like model, single-cut-or-join (SCJ), utilized the Fitch's algorithm to reconstruct ancestors based on binary characters in terms of gene adjacencies; however, the characters are not independent, since conflicting adjacencies cannot belong simultaneously to the same genome. The SCJ's strategy of solving the conflict is simply to initialize the root with absence thus any ambiguity state will be resolved at the root as absence. This method is regarded as the only known distance for which the ancestral genome reconstruction problem has a polynomial time solution [8].

The other type of model that handles gene adjacencies relies on two separate steps. First, the weight or probability that a gene adjacency is present in a genome is computed independently. Then those gene adjacencies are assembled into a valid ancestral genome. InferCAR [17] and its probabilistic version InferCAR-sPro are the pioneering methods based on this model. In this model, all combinations of gene adjacencies are considered, and their probabilities are computed by a variant of the Fitch's parsimony algorithm. Finally, a greedy heuristic is used for to assemble the genes into a valid genome.

Later by relaxing the constraint of gene adjacency to gapped adjacency, GapAdj is proposed with the computation of a rigorous score for each potential ancestral adjacency $(a,b)$, reflecting the maximum number of times $a$ and $b$ can be adjacent for any setting of ancestral genomes, as well as an algorithm to generate more reliable amount of chromosomes. Simulation experiments conducted by GapAdj show that GapAdj often ended up with a completely assembled genome, but resulted in a higher error rate that InferCAR.

# Algorithmic details

Our recent method *Probabilistic Method of Ancestral Genomics* (PMAG) [1] is also based on adjacencies and uses a probabilistic framework. It requires a given topology of the input genomes (assumed to be the phylogeny) and places the known genomes at the leaves. PMAG first encodes the gene orders into binary sequences and estimates the parameters in the transition model for

adjacency changes. It then checks each ancestral (internal) node in turn, and each will be computed independently by going through the following three steps: it first re-roots the input tree to have the target ancestor as the root of a new tree; it then uses a probabilistic inference tool to compute the conditional probabilities of all adjacencies; at last it uses a greedy algorithm similar to that presented in [17] to assemble a valid gene order. The last step is very critical, but the greedy algorithm tends to produce excessive number of contigs, indicating that it is very easy to get trapped in local optima. In this paper, we introduce a new assembly algorithm, which not only improves the accuracy of the assembled gene orders, but also reduce the number of contigs to be very close to the true result. The following are the details of our algorithm.

## Re-rooting the phylogeny

Before we can infer the ancestral genome of an internal node, we must first re-root the given phylogeny tree to that node, making it the root of the new tree, which is a standard procedure and has already been used in [1,9]. The underlying rationale is that the calculation of probabilities follows a bottom-up manner such that only the species in the sub-tree of the target node are considered, it will result in loss of information if the node is not the root. As we are dealing with binary trees, the re-rooting procedure will need some extra work to preserve the tree structure as demonstrated in Figure 1. In this figure, as the ancestral node we have interest in is genome $A1$, to re-root the tree on this genome, we have to add an auxiliary node $A1'$, but set the branch length between $A1$ and $A1'$ (dashed edge) as always 0.

## Obtaining probabilities of adjacencies

A gene order can be expressed as a sequence of adjacency information that specifies the presence or absence of all the adjacencies [18,19]. Denote the head of a gene $i$ by $i^h$ and its tail by $i^t$. We refer $+i$ as an indication of the direction from head to tail ($i^h \rightarrow i^t$) and otherwise $-i$ as ($i^t \rightarrow i^h$). We further write 1 (0) to indicate the presence (absence) of the adjacency and we consider only those adjacencies and telomeres that appear at least once in the input genomes.

Since we are handling binary sequences with two characters, we use a general time-reversible framework to simulate the transitions from presence (1) to absence (0) and vice versa. Since each genome contains $n+O(1)$ adjacencies and telomeres where $n$ is the gene number and $O(1)$ equals to the number of linear chromosomes in the genome, thus the probability that an adjacency changes from presence (1) to absence (0) in the sequence is $\frac{2}{n+O(1)}$ under one operation. Since there are up to $\binom{2n+2}{2}$ possible adjacencies and telomeres, the probability for an adjacency changing from absence (0) to presence (1) is $\frac{2}{2n^2+O(n)}$. Therefore, we come to the conclusion that the transition from 1 to 0 is roughly $2n$ times more likely than that from 0 to 1.

To show how the transition model and the re-rooting procedure can respectively influence the performance of PMAG, we compare PMAG to its three variants through simulations (see details later):

- Naive: The naive version of PMAG with a neutral model of adjacency changes and fixed tree topology for all ancestral nodes.
- Naive+Model: Naive method cooperating with the biased transition model.
- Naive+Re-rooting: Naive method cooperating with the re-rooting procedure.

Figure 2 summarizes the comparison result using the smaller datasets among the four methods with tree diameters from $1n$ (easy case) to $5n$ (very difficult). In general, higher tree diameter effectively increased the difficulties and hence reduced the portion of correct adjacencies all methods can recover. Unsurprisingly the Naive method is the least accurate in all cases, and both Naive+Model and Naive+Re-rooting can independently enhance the accuracy of Naive method. By incorporating both mechanisms, PMAG not only inherited both improvements, but also obtained additional improvements as well, suggesting the transition model and the re-rooting procedure be useful and indispensable for our method.

## Computing probabilities of gene adjacencies and assemble gene orders

Once we have the tree topology and binary sequences encoding the input gene orders, we use the extended probabilistic approach for sequence data described by Yang [20] to infer the ancestral gene orders at the root node, as described in detail in [1].

In the binary sequences, each site $k$ represents an adjacency with character either 0 (absence) or 1 (presence); for each site at the root node, we seek to calculate the conditional probability of observing that adjacency. Suppose $x$ is the root of a given tree, then the conditional probability that node $x$ has the character $s_x$ at site $k$, given $D_x$ representing the observed data at site $k$ in all

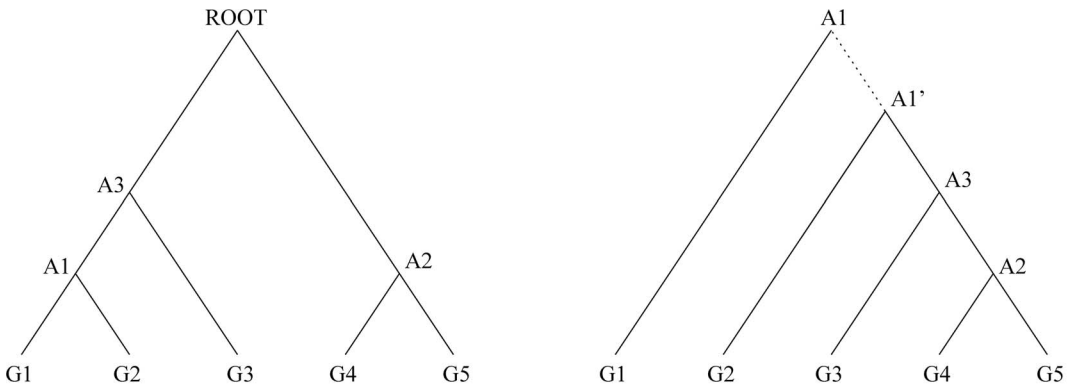


**Figure 1. Re-rooting the phylogeny tree from the original root to the ancestral node under inference which is $A1$ in this case.** Auxiliary node $A'$ is added to preserve its binary structure.
doi:10.1371/journal.pone.0108796.g001

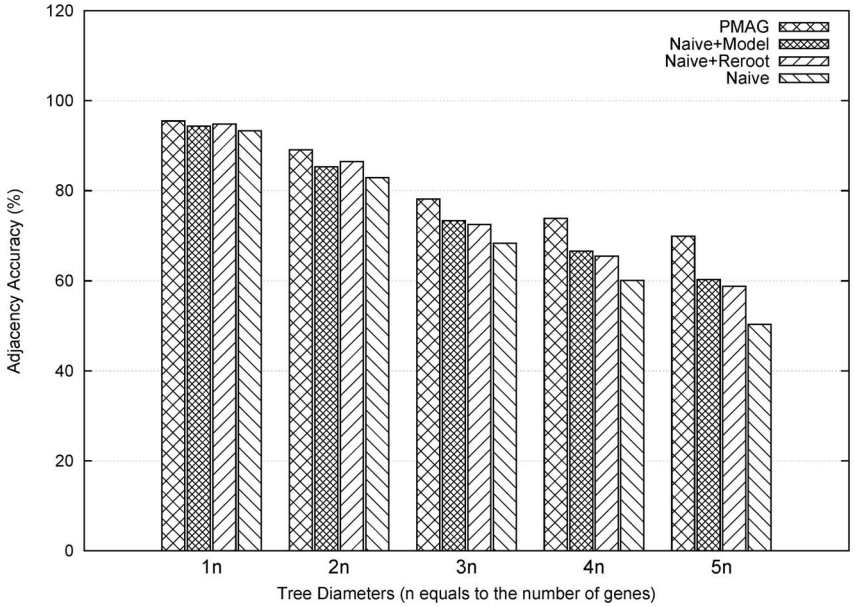


**Figure 2. Comparison of adjacency accuracy between PMAG and its three premature versions.** Datasets were simulated to have 10 genomes and 500 genes. X-axis represents the tree diameters from 1 to 5 times the number of genes.
doi:10.1371/journal.pone.0108796.g002

leaves of the sub-tree rooted at $x$, is

$$P(s_x|D_x)=\frac{P(s_x)P(D_x|s_x)}{P(D_x)}=\frac{\pi_{s_x}L_x(s_x)}{\sum_{s_x}\pi_{s_x}L_x(s_x)}$$

where $\pi_{s_x}$ is the character frequency for $s_x$.

For a site $k$, its conditional probability in the form of $L_x(s_x)$ is defined as the probability of observing the leaves that belong to the sub-tree rooted at $x$, given that the character of site $k$ at node $x$ is $s_x$. It can be calculated recursively in a post-order traversal fashion suggested by Felsenstein [1,21] as:

$$L_x(s_x)=\begin{cases}1 & \text{if } x \text{ is a leaf with character } = s_x \text{ at the site}\\ 0 & \text{if } x \text{ is a leaf with character } \neq s_x \text{ at the site}\\ \left[\sum_{s_f} p_{s_x s_f}(t_f)L_f(s_f)\right]\times\left[\sum_{s_g} p_{s_x s_g}(t_g)L_g(s_g)\right] & \text{otherwise}\end{cases}$$

where $f$ and $g$ are the two direct descendants of $x$. $p_{ij}(t)$ defines the transition probability that character $i$ changes to $j$ after an evolutionary distance $t$. As the true branch lengths are not available, we take advantage of the widely-used maximum-likelihood estimation from the binary sequences at the leaves to estimate the branch length.

Following the deduction of transition probability in [21], our transition-probability matrix can be written as

$$p_{ij}(t)=\pi_j+e^{-t}(\delta_{ij}-\pi_j)$$

Here the $\delta_{ij}$ is 1 if $i=j$, otherwise $\delta_{ij}$ is 0.

We use RAxML [22] as it has a method to handle binary sequences efficiently and follows the steps suggested by Yang [20]. We modified RAxML so that it takes into account the biased transition model.

Our improvement over [1] is a better algorithm to assemble gene adjacencies and telomere into a valid gene order, with the requirement that each gene appears exactly once in the ancestral genome. In general, a higher probability of the presence state implies an adjacency or telomere should be more likely to be included in the ancestor; however, the decision on choosing an adjacency or telomere cannot be solely made upon its own probability as each gene can only be selected once. In the original PMAG, ancestral adjacencies are assembled by the greedy heuristic based on the adjacency graph proposed by *Ma et al*. There are two issues with the greedy approach: 1) it can only achieve a good approximation for closely related genomes; 2) it tends to create new contigs instead of connecting genes, resulted in an excessive number of contigs. In this paper, we develop the following algorithm based on the observation from [23], i.e. it can transform the problem of obtaining gene orders from (conflict) adjacencies into an instance of Traveling Salesman Problem (TSP). Although the TSP is NP-hard, it is a widely studied problem with very good solvers exist.

Specifically, we will transform genes into cities and adjacency probabilities into edge weights, and our goal is to find a tour that traverses all genes with the largest combined probabilities along the tour. As most TSP solver aims at finding a tour with minimum cost, to use probabilities as edge weights, we convert them by taking their logarithmic values. Suppose for an ancestral node $K$ and a set of $m$ adjacencies $A=\{a_1,a_2,...,a_m\}$ and $n$ telomeres $T=\{t_1,t_2,...,t_n\}$ from leaf species, each with probabilities $P=\{p_{a_1},...,p_{a_m},p_{t_1},...,p_{t_n}\}$, we can create the TSP graph $G$ by first splitting each gene $g$ to two cities, denoted as $g^h$ and $g^t$ respectively, and representing each telomere $t$ by a unique vertex $e_i$, where $1\le i\le n$. To ensure a valid tour, we must connect $g^h$ and $g^t$ in a tour; thus we set the cost between $g^h$ and $g^t$ as $-\infty$. For any adjacency $(f,g)\in A$, we add an edge between $f^t$ and $g^h$; similar edges are added for other combinations of orientations $(-f,g)$, $(f,-g)$ and $(-f,-g)$, as well as genes connecting to telomeres. For the rest of edges, as we could not find a valid probability, it means these edges should have a very low chance to

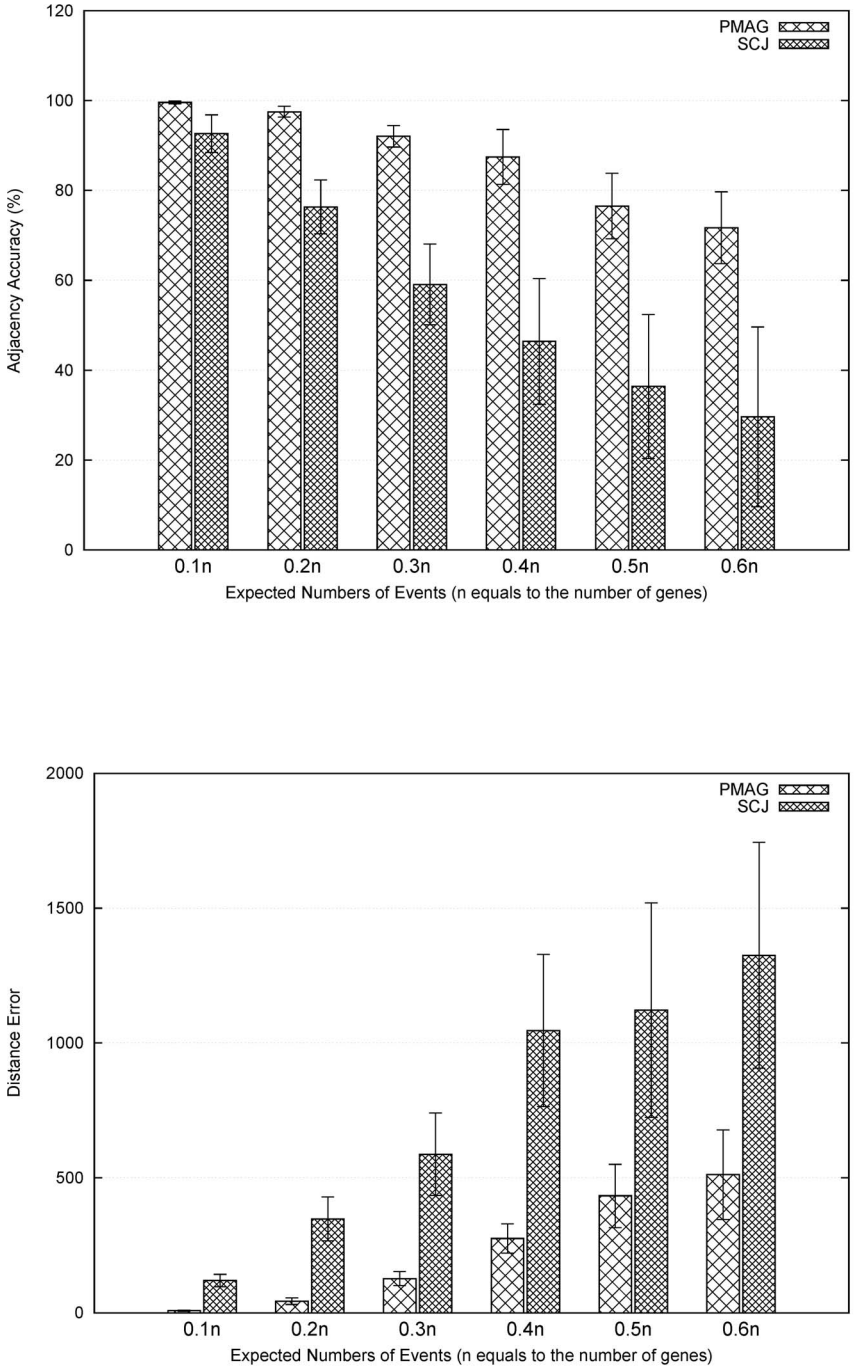


**Figure 3. Comparison of adjacency accuracy (top) and distance accuracy (bottom) between PMAG and SCJ.** Datasets were produced by the simulator provided in SCJ program that contain 32 genomes, each with five chromosomes and a total of 2,000 genes. X-axis represents the expected number of events from 0.1 to 0.6 times the number of genes.
doi:10.1371/journal.pone.0108796.g003

be present in the ancestral genome; thus we set the edge weights to $+\infty$ to exclude them from the solution. In the solution path, multiple contiguous caps are shrunk into a single one, and a gene segment between two caps is taken as a chromosome.

One of the best and most used TSP solver is Concorde [24], which we integrated it into MAG. For the solution path, multiple contiguous extremities are shrunk to a single one and a gene segment between two extremities is taken as a contig. Our construction of TSP topology is in a spirit similar to GapAdj, however GapAdj requires additional procedures and parameters to adjust the contig number. Instead, our inference of the ancestral genome is uniform and directly from the solution of TSP, minimizing the risk of introducing artifacts.

## Experimental Results

### Experimental design

Since actual ancestors are rarely known for sure, it is difficult to evaluate ancestral reconstruction methods with real datasets. In

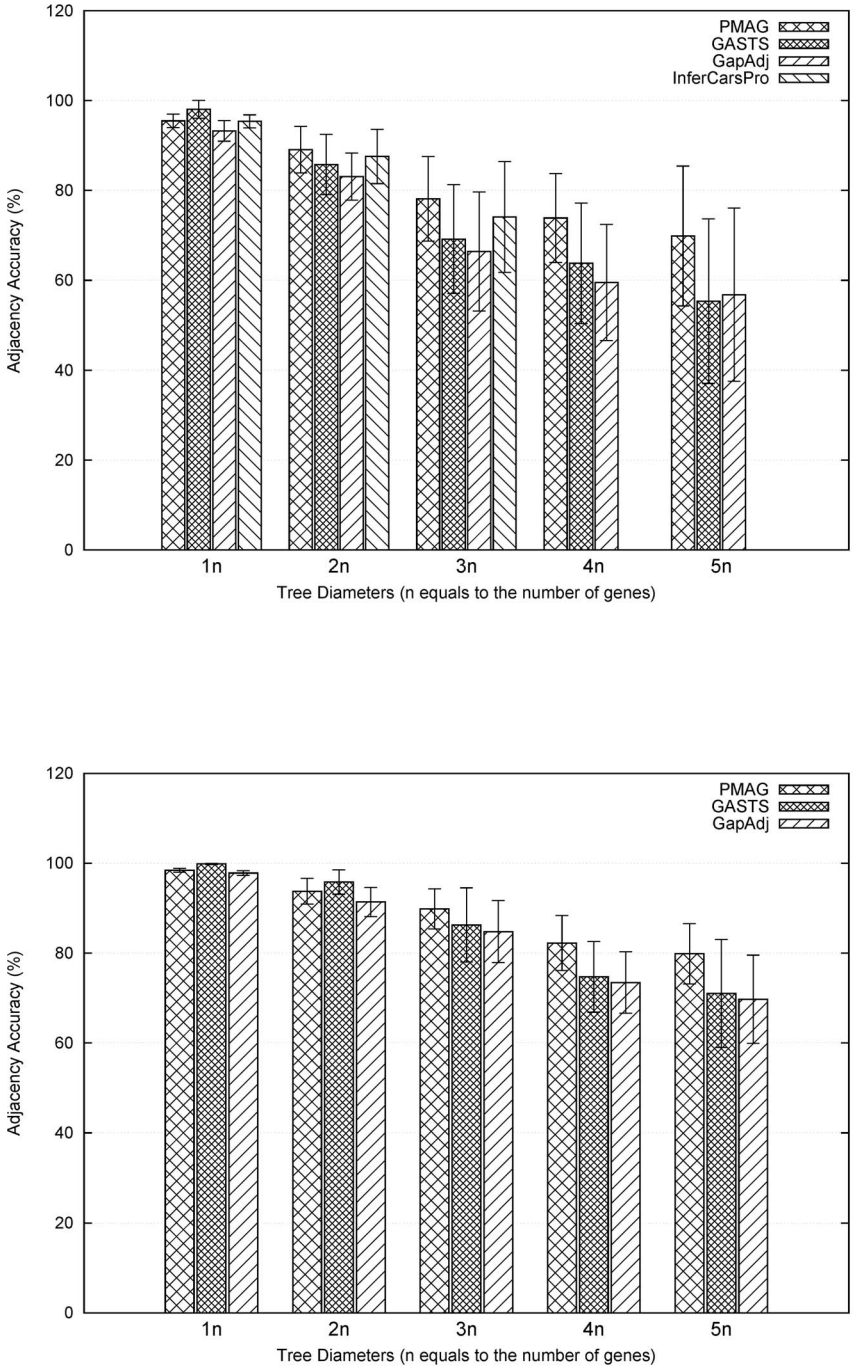


**Figure 4. Comparison of adjacency accuracy between PMAG, InferCARsPro, GASTS and GapAdj: (top) datasets contain 10 genomes and 500 genes; (bottom) datasets contain 20 genomes and 2000 genes.** Standard deviations are given at the top of bars. X-axis represents the tree diameters from 1 to 5 times the number of genes.
doi:10.1371/journal.pone.0108796.g004

order to carry out a complete evaluation over a group of methods under a wide range of configurations, we conducted a collection of simulation experiments following the standard steps of such tests that have been extensively adopted in genome rearrangement studies [19,25].

In particular, a group of tree topologies were first generated with respect to the expected tree diameters. An initial gene order was assigned at the root so it can evolve down to the leaves following the tree topology mimicking the natural process of evolution, by carrying out a number of predefined evolutionary events. In this way, we obtained the complete evolutionary history of the model tree and the whole set of genomes it has.

Normally we utilized the simulator proposed by Lin *et al.* [26] to produce birth-death tree topologies. Since SCJ has its own simulator, we used that simulator for a fair comparison in the tests involving SCJ. With a model tree, we can produce genomes of any size by simply adjusting four main parameters: the number of genomes $m$, the number of chromosome $c$, the number of genes $n$,

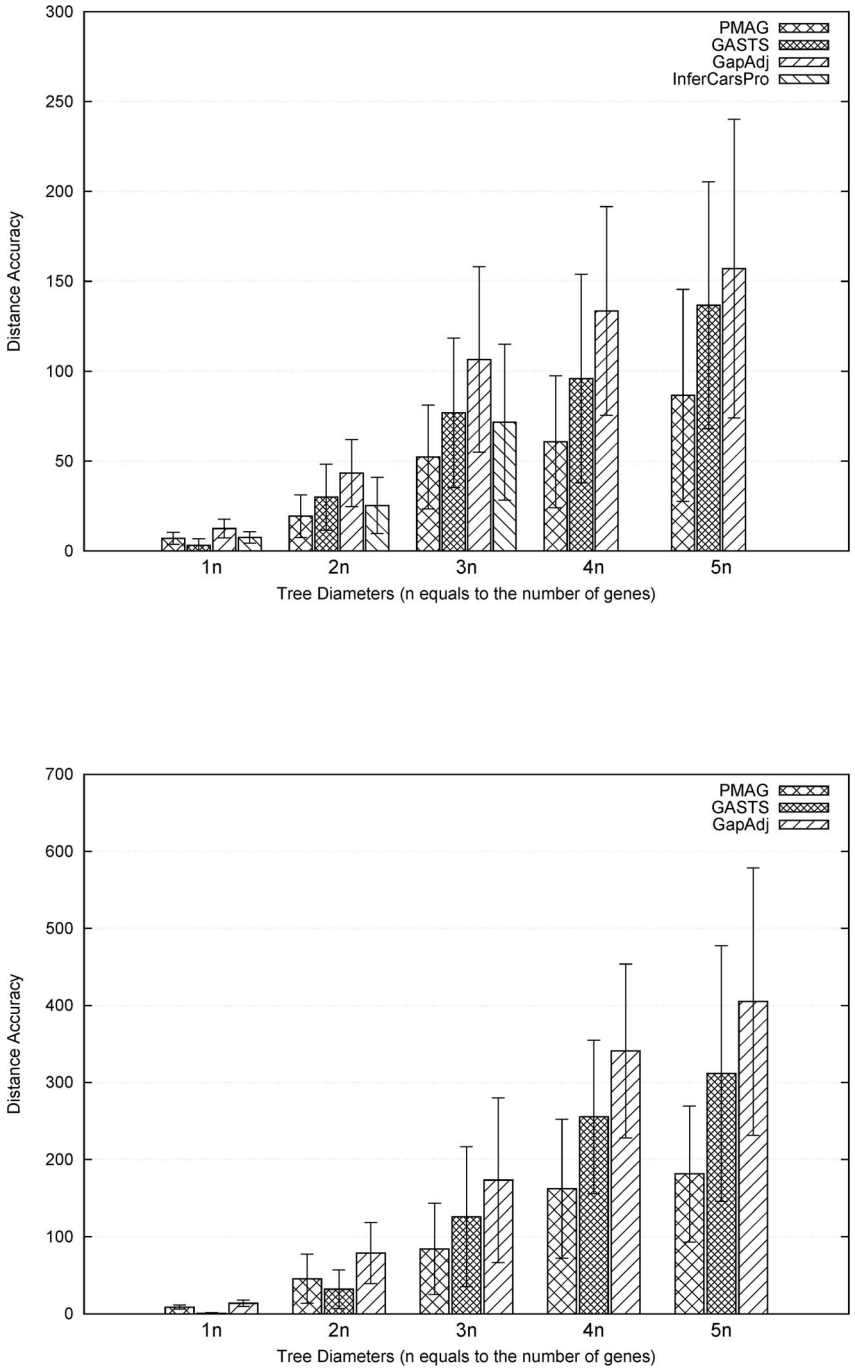


**Figure 5. Comparison of distance accuracy between PMAG, InferCARsPro, GASTS and GapAdj: (top) datasets contain 10 genomes and 500 genes; (bottom) datasets contain 20 genomes and 2000 genes.** Standard deviations are given at the top of bars. X-axis represents the tree diameters from 1 to 5 times the number of genes.
doi:10.1371/journal.pone.0108796.g005

and the tree diameter $d$ (equivalent to the branch length $l$ in SCJ's simulator). For the following simulation experiments, we generated datasets of two different sizes. First we generated a smaller datasets with 10 genomes, each with 500 genes and five chromosomes to closely mimic the rearrangement scenarios in bacterial genomes with multi-chromosomes. We also produced datasets of larger size contains 20 genomes, each with 2,000 genes and five chromosomes. Along each branch, we performed 80% random inversions and 20% random translocations to account for intra- and inter-chromosomal rearrangements.

The following three measurements were used to assess the predicted ancestral genomes. We first calculated the adjacency accuracy $C$ as the total number of correctly inferred adjacencies (i.e. those also appear in the true ancestral genomes) divided by the total number of adjacencies in both true genome and predicted genome. Second, we calculated distance accuracy $D$ defined as the DCJ distance between a predicted ancestor and its corresponding

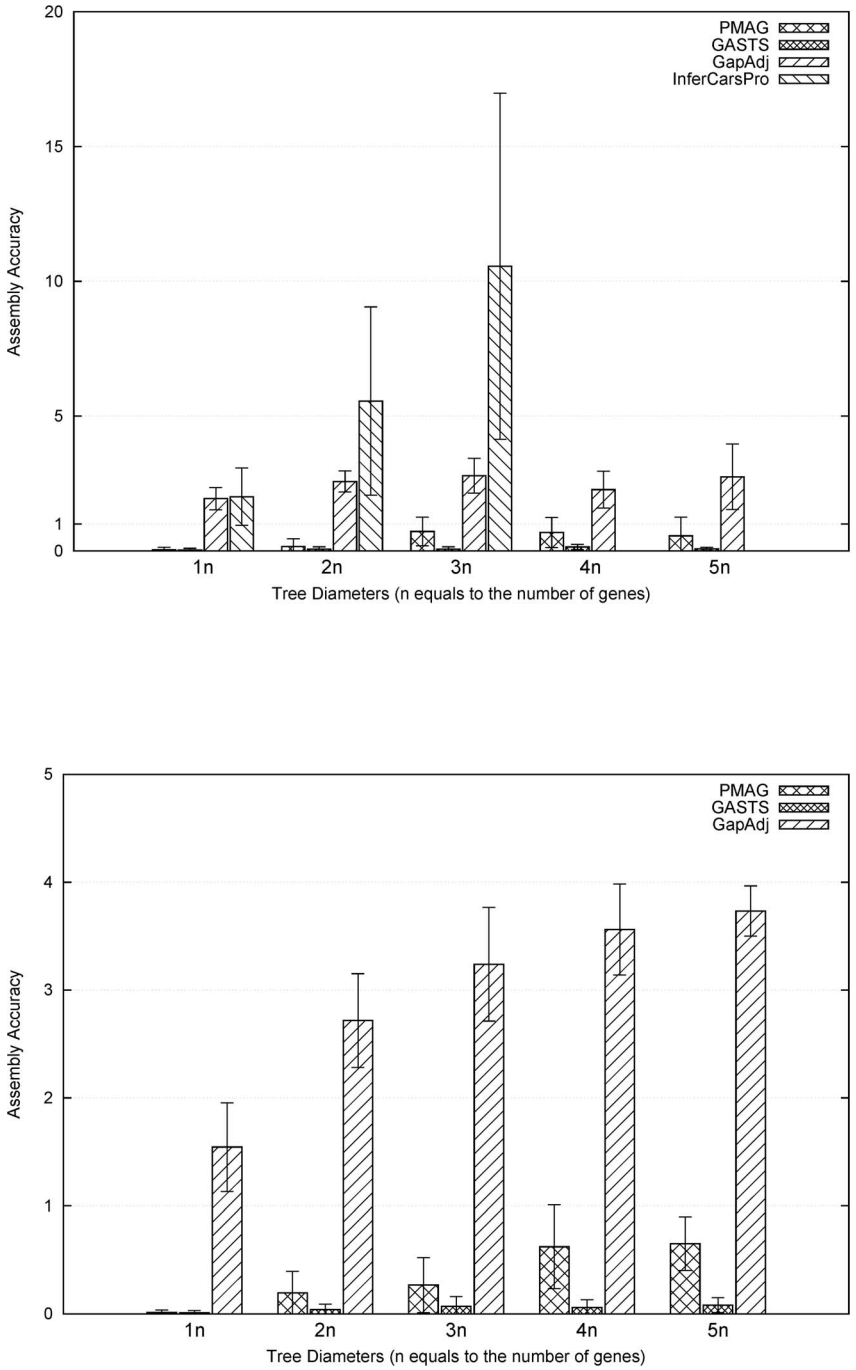


**Figure 6. Comparison of assembly accuracy between PMAG, InferCARsPro, GASTS and GapAdj: (top) datasets contain 10 genomes and 500 genes; (bottom) datasets contain 20 genomes and 2000 genes.** X-axis represents the tree diameters from 1 to 5 times the number of genes.
doi:10.1371/journal.pone.0108796.g006

**Table 1.** Comparison of assembly accuracy between PMAG and SCJ under different expected numbers of evolutionary events along a tree branch. (*n* equals to the number of genes).

| Tree Diameter | 0.1n | 0.2n | 0.3n | 0.4n | 0.5n | 0.6n |
|---|---|---|---|---|---|---|
| PMAG | 0 | 0.13 | 0.30 | 0.45 | 0.48 | 0.63 |
| SCJ | 157 | 476 | 785 | 1031 | 1280 | 1413 |

doi:10.1371/journal.pone.0108796.t001

**Table 2**. Time consumptions of four methods (including the previous greedy version of PMAG in analyzing large datasets under different tree diameters. ($n$ equals to the number of genes).

| Tree Diameter | PMAG | GapAdj | GASTS | PMAG-Greedy |
|---|---|---|---|---|
| 1n | 13 min | 10 min | 1 min | 1 min |
| 2n | 13 min | 12 min | 10 min | 3 min |
| 3n | 15 min | 12 min | 45 min | 3 min |
| 4n | 18 min | 14 min | 120 min | 4 min |
| 5n | 20 min | 16 min | 159 min | 5 min |

doi:10.1371/journal.pone.0108796.t002

true genome. Apparently for the genome rearrangement study, distance accuracy is more appropriate as it not only considers the adjacency changes, but also takes differences in genome structures into account. Finally, to assess the assembly capabilities, we computed assembly accuracy $A$ as the absolute differences of the number of chromosomes between a predicted ancestor and its corresponding truth. For each dataset, the average of each measurement across all ancestors was computed and for each tree diameter, we produced 10 datasets and reported their average, as well as their standard deviation.

## Evaluation of PMAG against SCJ

Simulator embedded in the SCJ was used, and the measurement of difficulty became the branch length $l$, denoting the expected number of evolutionary events along an edge of the tree which is sampled from an uniform distribution on the set $\{1,2,3...d\}$, where $d$ equals to $l \times n$ and $n$ is the number of genes. As before, those events consisted of 80% of inversions and 20% translocations. Since SCJ and PMAG are both fast enough, we therefore generated a set of larger dataset containing 32 genomes, each with five chromosomes and a total of 2,000 genes.

Figure 3 demonstrates the adjacency accuracy and the distance accuracy of PMAG and SCJ respectively. This figure clearly suggests that PMAG can significantly outperform SCJ in all tested cases.

## Evaluation of PMAG against other methods

In this section, we picked three main competitors from both event-based and adjacency-based methods, and compared them with PMAG. In particular we supplied InferCARsPro with multi-chromosomal genomic distances as its branch lengths computed by GRIMM [27]. Moreover, in GapAdj, the cutoff value and maximal iterations were set to 0.6 and 25 as suggested by the authors. The event-based method GASTS was simply run by providing the true tree and the input genomes. Results of InferCARsPro under large tree diameters were missing as it failed to finish the tests in three days.

Figure 4 shows the results measured by the adjacency accuracies. When the tree diameters were $1n$, all methods were able to produce highly accurate ancestral genomes ($>90\%$) and the differences among methods were not significant. In particular, GASTS was the most accurate method, while the performances of PMAG and InferCARsPro were similar, and both were better than GapAdj. As the tree diameters went larger, GASTS quickly became unreliable which is consistent with the experimental findings reported in the study of GASTS [7]. In all tests, PMAG showed great robustness against disturbance and achieved the highest adjacency accuracy when the tree diameter grows greater than $2n$. Figure 5 shows the results measured by the distance accuracies. In general, the relative performances of various methods in distance measurement are very similar to the adjacency accuracies.

## Comparison of performances on assembly

The final step of adjacency-based methods often involves assembly of adjacencies into contiguous segments which can be viewed as chromosomes or more precisely contigs. Previous methods InferCARsPro employing a greedy algorithm for assembly often ends up with an excessive number of contigs. Later the assembly accuracy was improved by GapAdj using the concept of gapped adjacencies with a sacrifice of accuracy.

Our measurement of accuracy only counts adjacencies correctly recovered. However, for two assembled gene orders with similar adjacency accuracy, the one with the number of contigs close to the number of chromosomes should be viewed as having better accuracy. Thus, we summarized the number of contigs produced by various methods and computed the averages of assembly accuracy for all cases in Figure 6. From the figure, the event-based method GASTS without the need for assembly of gene adjacencies produced the most relevant number of contigs. Among the adjacency-based methods, PMAG showed much better assembly performance, and its performance was very close to GASTS. As expected, the greedy assembly used in InferCARsPro produced the least relevant number of contigs. By examining Figure 4, we found that although PMAG returned more contigs than GASTS, its distance and adjacency accuracies were better, indicating that GASTS had a tendency to introduce bad adjacencies in order to keep the number of contigs small.

**Table 3.** Comparison of the adjacency accuracy between PMAG and its greedy version. The number of genomes is 20 and the number of genes is 2,000.

| Tree Diameter | 1n | 2n | 3n | 4n | 5n |
|---|---|---|---|---|---|
| PMAG | 98.7 | 93.5 | 89.7 | 82.2 | 79.5 |
| PMAG-Greedy | 98.5 | 93.2 | 88.6 | 80.2 | 77.8 |

doi:10.1371/journal.pone.0108796.t003

**Table 4.** Comparison of the assembly accuracy between PMAG and its greedy version. The number of genomes is 20 and the number of genes is 2,000.

| Diameter | 1n | 2n | 3n | 4n | 5n |
|---|---|---|---|---|---|
| PMAG | 0.03 | 0.2 | 0.3 | 0.6 | 0.6 |
| PMAG-Greedy | 2.1 | 5.5 | 8.5 | 15.9 | 17.6 |

doi:10.1371/journal.pone.0108796.t004

Table 1 shows the assembly accuracies of PMAG and SCJ. From the table, PMAG yielded very accurate amount of contigs; however, since SCJ is overly conservative, it missed a large portion of true adjacencies and produced a massive amount of contigs. In other words, SCJ has difficulty in assembling gene orders due to its overly simplified cost to weigh adjacencies.

### Time efficiency

All tests were conducted on a workstation with 2.4 GHz CPUs and 4 GB RAM. In general, SCJ is undoubtedly the fastest and can return results in just a few seconds. In the experiments with small datasets, InferCARsPro required the most amount of time and the other three methods can always finish within a minute.

We summarized the time consumptions of PMAG, GapAdj, GASTS and the previous greedy version of PMAG in handling large datasets in table 2. From the table, the running time of PMAG and GapAdj were very close and stable, and tree diameters did not remarkably slow down these programs. On the other hand, GASTS severely suffered from large tree diameters, suggesting its potential limitation in handling genomes that are distant to each other.

### Comparing PMAG with Its Previous Version

We compared PMAG with its greedy version to evaluate the new TSP approach. Table 3 and Table 4 showed the adjacency and assembly accuracy, respectively. These tables suggested that although the TSP solver was about 10 times slower than the greedy solver (Table 2), the new PMAG method had achieved improved adjacency accuracy with much better performance in term of the number of recovered contigs.

## Conclusions and Future Work

In this paper, we introduced the adjacency-based method PMAG in the probabilistic framework for ancestral gene-order inference. PMAG determines the state for each adjacency in the binary encoding to be either present or absent in an ancestral genome according to its conditional probability. Ancestral gene orders are finally assemblies by connecting individual adjacencies into continuous regions using a TSP approach Experimental results reveal that PMAG can not only accurately infer ancestral genomes, and also did a good job in assembling adjacencies into valid gene orders. Finally, PMAG is fast and also stable across a wide range of configurations.

However, much work remains to be done. As PMAG relies on gene adjacencies, how to recover adjacencies lost in evolution (thus not shown in leave genomes) is an interesting problem. In the current implementation, these adjacencies have no definite edge weight and they are all set as $+\infty$. As a result, the TSP tour is prevented from passing through them, although they may be better choices. Our experiments showed that these adjacencies account for about 25% of errors in PMAG; thus, we need to devise a method that can assign better edge weights to missing adjacencies. Since each internal node can be computed independently, the speed of PMAG can be further improved by utilizing the presence of multiple computing cores in modern CPUs by placing each node's computation on a core.

## Author Contributions

Conceived and designed the experiments: NY FH JT. Performed the experiments: NY FH LZ. Analyzed the data: NY FH JT LZ. Contributed reagents/materials/analysis tools: NY FH. Wrote the paper: NY FH JT.

## References


1. Hu F, Zhou L, Tang J (2011) Reconstructing Ancestral Genomic Orders Using Binary Encoding and Probabilistic Models. Proceedings of the 9th International Symposium on Bioinformatics Research and Applications (ISBRA), 17–27.
2. Kent W, Baertsch R, Hinrichs A, Miller W, Haussler D (2003) Evolutions cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. Proceedings of the National Academy of Sciences, 100(20):11484–11489.
3. Muller K, Borsch T, Legendre L, Porembski S, Theisen I, et al. (2008) Evolution of carnivory in Lentibulariaceae and the Lamiales. Plant Biology, 6(4):477–490.
4. Moret BME, Wyman S, Bader DA, Warnow T, Yan M (2001) A New Implmentation and Detailed Study of Breakpoint Analysis. Proceedings of the 6th Pacific Symposium on Biocomputing (PSB), 583–594.
5. Bourque G, Pevzner P (2002) Genome-scale evolution: reconstructing gene orders in the ancestral species. Genome Research, 12(1): 26–36.
6. Alexeyev M, Pevzner P (2009) Breakpoint graphs and ancestral genome reconstructions. Genome Research, 19 (5): 943–957.
7. Xu W, Moret BME (2011) GASTS: Parsimony scoring under rearrangements. Proceedings of the 11th Workshops on Algorithms in Bioinformatics (WABI), 351–363.
8. Biller P, Feijao P, Meidanis J (2012) Rearrangement-based phylogeny using the Single-Cut-or-Join operation. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 10 (1), 122–134.
9. Ma J (2010) A probabilistic framework for inferring ancestral genomic orders. Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 179–184.
10. Gagnon Y, Blanchette M, El-Mabrouk N (2012) A flexible ancestral genome reconstruction method based on gapped adjacencies. BMC bioinformatics 13 (Suppl 19): S4.
11. Yancopoulos S, Attie O, Friedberg R (2005) Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics, 21(16): 3340–3346.
12. Feijao P, Meidanis J (2009) SCJ: a variant of breakpoint distance for which sorting, genome median and genome halving problems are easy. Proceedings of the 9th Workshop on Algorithms in Bioinformatics (WABI), 85–96.
13. Caprara A (1999) Formulations and hardness of multiple sorting by reversals. Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB), 84–93.
14. Tannier E, Zheng C, Sankoff D (2009) Multichromosomal median and halving problems under different genomic distances. BMC Bioinformatics, 10(1): 120.
15. Xu W, Sankoff D (2008) Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. Proceedings of the 8th Workshop on Algorithms in Bioinformatics (WABI), 25–37.
16. Hannenhalli S, Pevzner P (1995) Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. Proceedings of the 27th Annual ACM Symposium on Theory of Computing, 178–189.
17. Ma J, Zhang L, Suh B, Raney B, Burhans R, et al. (2006) Reconstructing contiguous regions of an ancestral genome. Genome Research, 16(12):1557–1565.
18. Hu F, Gao N, Tang J (2011) Maximum likelihood phylogenetic reconstruction using gene order encodings. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 117–122.

19. Lin Y, Hu F, Tang J, Moret BME (2013) Maximum likelihood phylogenetic reconstruction from high-resolution whole-genome data and a tree of 68 eukaryotes. Proceedings of the 18th Pacific Symp. on Biocomputing (PSB), 285–296.
20. Yang Z, Kumar S, Nei M (1995) A new method of inference of ancestral nucleotide and amino acid sequences. Genetics, 141(4):1641–1650.
21. Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution, 17(6):368–376.
22. Stamatakis A (2006) RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics, 22(21):2688–2690.
23. Tang J, Wang L (2005) Improving genome rearrangement phylogeny using sequence-style parsimony. Proceedings of 5th IEEE Symposium on Bioinformatics and Bioengineering, 137–144.
24. Applegate D, Bixby R, Chvatal V, Cook W (2003) Concorde TSP solver. Available: http://www.tsp.gatech.edu/concorde. Accessed September 4th 2014.
25. Jahn K, Zheng C, Kovac J, Sankoff D (2012) A consolidation algorithm for genomes fractionated after higher order polyploidization. BMC Bioinformatics, 13 (Suppl 19): S8.
26. Lin Y, Rajan V, Moret BME (2011) Fast and accurate phylogenetic reconstruction from high-resolution whole-genome data and a novel robustness estimator. Journal of Computational Biology, 18(9): 1131–1139.
27. Tesler G (2002) Efficient algorithms for multichromosomal genome rearrangements. Journal of Computer and System Sciences, 65(3): 587–609.